Unleashing the Potential of Text-attributed Graphs: Automatic Relation Decomposition via Large Language Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

Recent advancements in text-attributed graphs (TAGs) have significantly improved the quality of node features by using the textual modeling capabilities of language models. Despite this success, utilizing text attributes to enhance the predefined graph structure remains largely unexplored. Our extensive analysis reveals that conventional edges on TAGs, treated as a single relation (*e.g.*, hyperlinks) in previous literature, actually encompass mixed semantics (*e.g.*, "advised by" and "participates in"). This simplification hinders the representation learning process of Graph Neural Networks (GNNs) on downstream tasks, even when integrated with advanced node features. In contrast, we discover that decomposing these edges into distinct semantic relations significantly enhances the performance of GNNs. Despite this, manually identifying and labeling of edges to corresponding semantic relations is labor-intensive, often requiring domain expertise. To this end, we introduce **RoSE** (**R**elation-**o**riented **S**emantic **E**dge-decomposition), a novel framework that leverages the capability of Large Language Models (LLMs) to decompose the graph structure by analyzing raw text attributes - in a *fully automated* manner. **RoSE** operates in two stages: (1) identifying meaningful relations using an LLM-based generator and discriminator, and (2) categorizing each edge into corresponding relations by analyzing textual contents associated with connected nodes via an LLM-based decomposer. Extensive experiments demonstrate that our model-agnostic framework significantly enhances node classification performance across various datasets, with improvements of up to 16% on the Wisconsin dataset.

1 Introduction

Text-attributed graphs (TAGs) [1], which combine graph structures with textual data, are frequently used in diverse real-world applications, including fact verification [2; 3], recommendation systems [4], and social media analysis [5]. In TAGs, texts are incorporated as node descriptions such as paper abstracts in citation networks [6; 7; 8] or web page contents in hyperlink networks [9; 10]. By leveraging the rich information present in both the graph topology and its associated text attributes, substantial advancements have been achieved in graph representation learning. Among them, numerous studies have been proposed to enhance the node representation quality of TAGs by leveraging features generated from light-weighted pre-trained language models (PLMs) [1; 11; 12; 13; 14; 15] such as Sentence-BERT [16], or by refining raw texts using the general knowledge of Large Language Models (LLMs) [17; 15].

Despite their success, the potential of utilizing text attributes to enhance the predefined *graph structure* remains largely under-explored. Existing approaches have treated the edges in TAGs as a uniform relation, overlooking the diverse inherent semantics they convey. For instance, in the WebKB dataset [10], nodes denote web pages with their textual content as node features while their edges are formed by hyperlinks. Despite the presence of varying semantic meanings such as "node A is advised by node B" or "node A participates in node C", the relationships are bundled as a single relation type ("hyperlinks"), inadvertently entangling their semantic meanings. Such an over-simplification limits the ability of Graph Neural Networks (GNNs) to accurately model the intricate relationships between nodes, resulting in suboptimal performance.

Throughout our comprehensive analysis, we reveal that the downstream task performance of GNNs is hindered by the oversimplified graph structure, even when integrating node features obtained from PLMs. On the other hand, disentangling edges into multiple semantic types yields more distinguishable representations that significantly enhance downstream performance. However, manually identifying and labeling relation types is labor-intensive as it requires human annotation and often necessitates domain expertise to determine meaningful relation types.

To address these challenges, we propose **RoSE** (**R**elation-**o**riented **S**emantic **E**dge-decomposition), a novel framework that utilizes LLMs to decompose predefined edges into semantic relations via textual information in a *fully-automated* manner. Given the description of the original graph composition, **RoSE** carefully identifies a concise set of meaningful relation types through the interaction between an LLM-based generator and a discriminator. Subsequently, the LLM-based decomposer disentangles each edge into predefined relation types by analyzing raw textual contents associated with its connected nodes. The versatility of our proposed framework is readily extended to varying architectures, encompassing edge-featured GNNs [18; 19; 20] and multi-relational GNNs [21; 22; 23].

Our contributions are summarized as follows:

- We reveal that the oversimplified graph structure in TAGs hinders the performance of GNNs on downstream tasks despite the integration of informative node features. On the other hand, mitigation through decomposing graph edges lead to significant enhancements in GNN performance.
- We present **RoSE**, a novel edge decomposition framework that utilizes the general reasoning capability of LLMs. **RoSE** identifies semantic relations through the interaction between an LLM-based generator and discriminator, and categorizes each edge into these relation types by analyzing textual contents via LLM-based decomposer. All these processes are automated, eliminating the need for extensive human analysis and annotation.
- Extensive evaluations on diverse TAGs and GNN architectures demonstrate the effectiveness of **RoSE** in improving node classification performance. Notably, our framework achieves improvements of up to 16% on the Wisconsin dataset.

2 Preliminaries

Node Classification with Graph Neural Networks. We study a TAG $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$, comprising N nodes in $\mathcal{V}$ along with a node-wise text attribute $\mathcal{T} = \{t_i | i \in \mathcal{V}\}$ and $M = |\mathcal{E}|$ undirected edges connecting nodes. Nodes are characterized by a feature matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N]^\mathsf{T} = g_{\boldsymbol{\phi}}(\mathcal{T}) \in \mathbb{R}^{N \times F}$, where their text attributes are encoded using a PLM $g_{\boldsymbol{\phi}}$ which is typically frozen. Edges are described by a binary adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{N \times N}$, with $\boldsymbol{A}[i, j] = 1$ if an edge $(i, j) \in \mathcal{E}$, and $\boldsymbol{A}[i, j] = 0$ otherwise.

Our focus lies on a node classification task using a GNN $f_{\boldsymbol{\theta}}$. The GNN learns representation of each node i by iteratively aggregating representations of its neighbors in the neighborhood set $\mathcal{N}_i$ in the previous layer, formulated as:

$$\boldsymbol{h}_i^{(l+1)} = \psi\big(\boldsymbol{h}_i^{(l)},\ \mathtt{AGG}(\{\boldsymbol{h}_j^{(l)}, \forall j \in \mathcal{N}_i\})\big). \tag{1}$$

Here, AGG denotes an aggregation function and ψ combines the node's prior representation with that of its aggregated neighbors. The initial representation is $\boldsymbol{h}_i^{(0)} = \boldsymbol{x}_i$ for notational simplicity and the overall multi-layered process can be expressed as $f_{\boldsymbol{\theta}}(\boldsymbol{X}, \boldsymbol{A})$. The objective function $\mathcal{L}$ used for training the GNN is defined as the cross-entropy loss between the predicted class probabilities $\boldsymbol{P} = \text{Softmax}(\boldsymbol{Z}) = \text{Softmax}\big(f_{\boldsymbol{\theta}}(\boldsymbol{X}, \boldsymbol{A})\big) \in \mathbb{R}^{N \times K}$ and the ground-truth labels $\boldsymbol{Y} \in \mathbb{R}^{N \times K}$:

$$\mathcal{L}_{\boldsymbol{\theta}} = -\frac{1}{N} \sum_{i \in \mathcal{V}}^{N} \sum_{k=1}^{K} \boldsymbol{Y}_{ik} \log \boldsymbol{P}_{ik}, \tag{2}$$

where $\boldsymbol{Z}$ represents the logit produced by the GNN and K represents the total number of classes.

Table 1: Node classification accuracy (%) on WebKB and IMDB datasets, trained with single and multi-type relations, averaged over 10 runs (± SEM). The best performances are represented by **bold**.

Datasets		Cornell	Texas	Wisconsin	IMDB
RGCN	Single Type	57.60 ± 1.78	65.88 ± 1.86	59.22 ± 1.70	62.96 ± 0.44
	Multi Type	**68.80 ± 1.88**	**76.47 ± 1.82**	**83.28 ± 1.64**	**68.66 ± 0.57**
HAN	Single Type	56.00 ± 1.67	68.82 ± 2.12	58.28 ± 1.99	63.24 ± 0.54
	Multi Type	**60.40 ± 1.91**	**71.37 ± 2.24**	**76.09 ± 1.88**	**68.39 ± 0.62**

Prompting Large Language Models. LLMs pre-trained on a vast amount of text corpora have demonstrated remarkable general reasoning capabilities proportional to their number of parameters [24; 25; 26; 27]. This advancement has led to a new approach to task alignment, allowing for the direct output obtainment from natural language prompts without the need for additional fine-tuning [28; 29; 30]. In practice, a natural language text prompt $\boldsymbol{s}$ is concatenated with a given input sequence $\boldsymbol{q} = \{q_i\}_{i=1}^{n}$ to form a new sequence $\widetilde{\boldsymbol{q}} = \{\boldsymbol{s}\} \cup \boldsymbol{q}$. Subsequently, an LLM $\mathcal{M}$ receives $\widetilde{\boldsymbol{q}}$ as its input and generates an output comprising a sequence of tokens $\boldsymbol{a} = \{a_i\}_{i=1}^{m} = \mathcal{M}(\widetilde{\boldsymbol{q}})$.

3 Analysis: Uncovering the Importance of Semantic Edge Decomposition

In this section, we analyze the potential performance improvements of GNNs when applied to TAGs with available semantic edge types. Toward this, we choose three TAG datasets of a small size enough to manually classify the semantic types of edges. First, we perform our analysis on WebKB hyperlink graphs (Cornell, Texas, Wisconsin) [10], where nodes represent web pages and edges indicate hyperlinks between nodes. Despite traditionally being treated as single relation graphs, their edges can be mainly categorized into multiple semantic types, such as "participates in", "advises/advised by", "being part of", and "supervised by". To the best of our knowledge, this is the first analysis to broadly create and label relation types in such graphs to verify GNNs' performance in a multi-relational scenario. Additionally, we include the IMDB graph [31], which consists of movie nodes with edges reflecting overlaps between movie professionals. In contrast to the WebKB graphs, the edges in the IMDB graph have been consistently regarded as multi-relations [22; 32], differentiated into "actor/actress overlap" and "director overlap". By incorporating this dataset into our analysis, we demonstrate the potential performance degradation when inherent relations are simplified as a single relation.

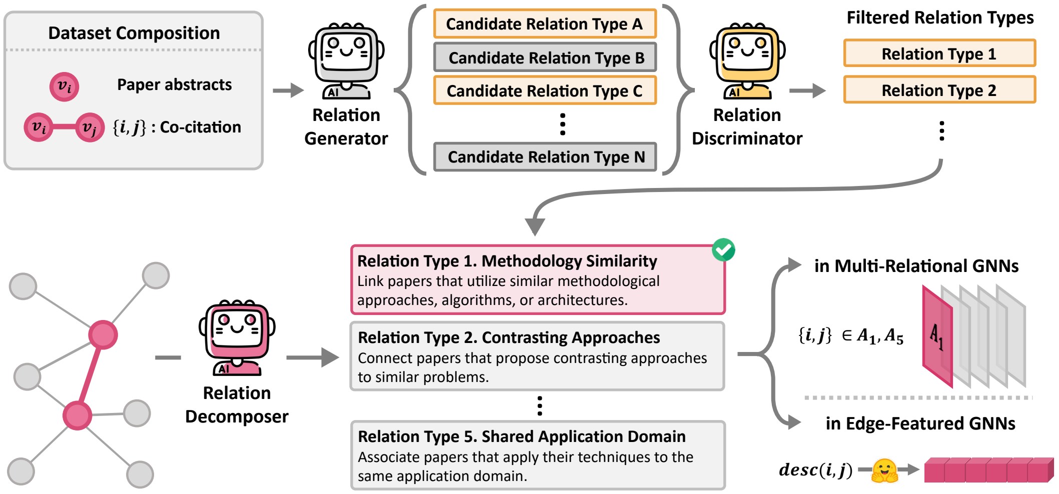
We evaluate the efficacy of relation labeling under the node classification task, with two multi-relational GNN architectures; namely RGCN [21] and HAN[1] [22]. Each is an extension of GCN [33] and GAT [34] to multi-relational scenarios, equipped with an edge type-specific neighborhood aggregation scheme (detailed formulation is outlined in Section 4.3). Note that in the case of training with a single relation, RGCN and HAN function similarly to asymmetric GCN and GAT, correspondingly. We train these GNNs in two different approaches: processing edges as a single and multiple types of relation.

As demonstrated in Table 1, decomposing edges into multiple semantic relations leads to significant performance improvements across all datasets and GNN architectures. This enhancement is particularly pronounced in the Wisconsin dataset, where accuracy improvements of 26.56% and 19.37% are achieved for RGCN and HAN, respectively. Furthermore, our analysis reveals that neglecting the entangled semantics in multi-relational benchmark results in suboptimal performance. The benefits of decomposition are also evident at the representation level, showing more distinguishable and clustered node representations, as illustrated in Figure 3 and 4 in Appendix B. Hence, our observation highlights the suboptimality present within the graph structure due to its oversimplification of edges, which can be adequately addressed through the decomposition of edges into distinct semantic relations.

4 RoSE: Relation-oriented Semantic Edge-decomposition

Despite the efficacy of semantic edge decomposition introduced in Section 3, the practical implementation of semantic edge decomposition presents several challenges. To begin with, defining

[1] Due to the scope of our research on semantic edge decomposition, we do not consider node type-wise aggregation in HAN.



Relation Type 1. Methodology Similarity ✓
Link papers that utilize similar methodological approaches, algorithms, or architectures.

Relation Type 2. Contrasting Approaches
Connect papers that propose contrasting approaches to similar problems.

⋮

Relation Type 5. Shared Application Domain
Associate papers that apply their techniques to the same application domain.

Figure 1: Overall framework of **RoSE**.

the appropriate semantic relation type is a non-trivial task that often requiring domain expertise. Moreover, creating annotations for the numerous edge types is extremely labor-intensive. In turn, limiting the usage of fine-tuned PLMs for edge decomposition, as they necessitate the identified list of edge types and the ground-truth edge labels for fine-tuning.

To address this, we present **RoSE**, an innovative framework that leverages the advanced textual reasoning capabilities of LLMs to automate the decomposition of edges into their inherent semantic relations based on their corresponding text attributes. **RoSE** is structured into two main phases: (1) Relation Type Identification (Section 4.1), and (2) Semantic Edge Decomposition (Section 4.2). The edges decomposed by **RoSE** can be seamlessly integrated with conventional GNN architectures in a plug-and-play manner (Section 4.3). This is facilitated either through direct edge type-specific neighborhood aggregation in multi-relational GNNs or by assigning relation types as edge features in edge-featured GNNs. In addition, to enhance efficiency, we introduce an edge sampling strategy that reduces the number of queries required for LLM-based edge type annotation (Section 4.4). Figure 1 illustrates the overall framework of **RoSE**.

4.1 Relation Type Identification

To decompose each edge into underlying semantic relations, it is essential to identify relation types that are: (1) meaningful, capturing the inherent context of predefined edges; (2) feasible, determinable based solely on textual attributes; and (3) distinct, ensuring clarity and avoiding redundancy within the graph.

We use a combination of an LLM-based *relation generator* and *relation discriminator* for this task. The *relation generator* addresses the requirement for meaningfulness by generating a set of plausible candidate relations based on graph composition. The *relation discriminator* ensures feasibility and distinctiveness by filtering out candidate relation types that exceed the analytical capability of LLMs or exhibit excessive redundancy. The effectiveness of this generator - discriminator framework is outlined in Section 5. We provide detailed information of each component in the following paragraphs. All prompt templates fixed throughout our experiments is specified in Appendix A.

Relation Generator. To obtain a set of edge types relevant to the given graph, we provide the *relation generator* $\mathcal{M}_g$ with detailed information about the graph in the input prompt s_g, which is mathematically formulated as $\mathcal{M}_g(s_g)$. This information includes specifying node's textual attributes (*e.g.*, paper abstracts), predefined rules for node connectivity (*e.g.*, co-citation), and category names (*e.g.*, rule learning). Subsequently, we outline the role of $\mathcal{M}_g$ and specifies the preliminary requirements for identifying meaningful relations within the graph. Based on the provided graph composition and task description, the *relation generator* generates a list of candidate relation types in a zero-shot manner, without any additional fine-tuning.

Relation Discriminator. To ensure the feasibility and distinctiveness of the generated relation types, we employ a *relation discriminator* $\mathcal{M}_d$. The discriminator $\mathcal{M}_d$ takes the relation types generated by $\mathcal{M}_g$ as input and filters out those that are irrelevant or infeasible to infer given the textual attributes and the analytical capabilities of LLMs. Given the set of candidate relation types output $\mathcal{M}_g(\boldsymbol{s}_g)$ by prompting *relation generator*, we concatenate $\mathcal{M}_g(\boldsymbol{s}_g)$ with the task description prompt $\boldsymbol{s}_d$ and pass the combined prompt to the *relation discriminator*.

The overall process can be formulated as obtaining a relation set $\mathbf{R} = \{\mathcal{R}_1, \mathcal{R}_2, ..., \mathcal{R}_R\}$ from the two-stage LLM outputs, represented as $\mathcal{M}_d\big(\{\boldsymbol{s}_d\} \cup \mathcal{M}_g(\boldsymbol{s}_g)\big)$, where $\mathcal{R}_r$ represents the textual description of r-th semantic relation. It is worth noting that in certain scenarios, there could be domain experts who can define the relation types with minimal cost. In such cases, the above process can be considered optional, as the predefined relation types can be directly fed to the LLM for edge decomposition. However, in the absence of domain expertise, our identification framework provides an automated and scalable solution.

4.2 Semantic Edge Decomposition

Given the set of semantic relation types $\mathbf{R}$ identified in Section 4.1, we deploy an LLM-based *relation decomposer* $\mathcal{M}_c$ tasked with assigning relevant relations to each edge (i, j). A major advantage of utilizing LLMs in this context is their capability to perform multi-label classification, useful in realistic scenarios where a single edge often convey multiple semantic meanings. For instance, in an IMDB graph, two connected movie nodes might share both a common director and actor. Reflecting such real-world complexities, we instruct $\mathcal{M}_c$ to determine all possible relations that the given edge can be categorized under. Equipped with raw texts t_i and t_j associated with nodes v_i and v_j, the decomposition process is expressed as $\mathcal{M}_c\big(\{\boldsymbol{s}_c\} \cup \{t_i, t_j\}\big)$ with $\boldsymbol{s}_c$ indicating the instruction prompt for $\mathcal{M}_c$.

4.3 Integration with Conventional GNNs

The edges disentangled by the *relation decomposer* can be flexibly integrated into either multi-relational GNNs [21; 22; 23] or edge-featured GNNs [18; 19; 20], highlighting its versatility.

Multi-Relational GNNs. When paired with multi-relational GNNs, the decomposed edges categorized into R types of relations are treated as R distinct sub-structures $\{\mathcal{E}_1, \mathcal{E}_2, ..., \mathcal{E}_R\}$. When a single edge is assigned with multiple relation types, it is included in several corresponding $\mathcal{E}_r$. Each set $\mathcal{E}_r$ is utilized to perform type-specific neighborhood aggregation. For a given node i at the l-th layer, these multi-relational GNNs are mathematically formulated as follows:

$$\boldsymbol{h}_i^{(l+1)} = \psi_{\text{rel}}\Bigg(\boldsymbol{h}_i^{(l)},\ \big\{\texttt{AGG}(\{\boldsymbol{h}_j^{(l)}, \forall j \in \mathcal{N}_i^{(r)}\})\big\}_{r=1}^{R}\Bigg), \tag{3}$$

where $\mathcal{N}_v^{(r)}$ denotes the set of neighbors of v connected via type-r relation. Here, ψ_{rel} represents the update function that combines outputs from edge type-wise aggregation (and optionally, the hidden representation of itself [21]). In general, ψ_{rel} is implemented using mean, (weighted) sum, or attention operators.

Edge-Featured GNNs. In addition, the decomposed edges facilitated by **RoSE** can be incorporated as edge features for edge-featured GNNs. Specifically, given relation type descriptions $\mathbf{R} = \{\mathcal{R}_1, \mathcal{R}_2, ..., \mathcal{R}_R\}$ curated from *relation generator* and *discriminator*, we utilize the same PLM g_ϕ employed for encoding node features to embed each type description $\mathcal{R}_r$, yielding a set of relational features. Subsequently, for each edge (i, j), the edge feature $\boldsymbol{e}_{ij}$ is assigned as the relational feature corresponding to the specific relation type associated with that edge, as determined by the *relation decomposer*. In cases where multiple edge types are applicable to a single edge, we incorporate all relevant edge features by duplicating the edge with each corresponding type. The operations for an individual node i at the l-th layer in edge-featured GNNs are formulated as follows:

$$\boldsymbol{h}_i^{(l+1)} = \psi\Bigg(\boldsymbol{h}_i^{(l)},\ \texttt{AGG}\big(\{\boldsymbol{h}_j^{(l)}, \xi^{(l+1)}(\boldsymbol{e}_{ij}) | \forall j \in \mathcal{N}_i\}\big)\Bigg), \tag{4}$$

where $\xi^{(l+1)}$ denotes a function that linearly maps $\boldsymbol{e}_{uv}$ to the same representational space as $\boldsymbol{h}_u^{(l)}$.

Table 2: Node classification accuracy (%) on various datasets and GNN architectures, averaged over 10 runs (± SEM). The best and second best performances are represented by **bold** and $\underline{\text{underline}}$.

Type	Model	Pubmed	IMDB	Cornell	Texas	Wisconsin	Cora	WikiCS	Avg Gain
Single-type	**GCN**	89.32 ± 0.11	64.04 ± 0.43	48.20 ± 2.18	62.94 ± 2.49	51.56 ± 1.79	88.05 ± 0.40	82.58 ± 0.27	-
	GAT	88.64 ± 0.11	64.39 ± 0.44	57.00 ± 1.56	66.86 ± 1.48	56.25 ± 2.29	87.74 ± 0.38	82.79 ± 0.16	-
	JKNet	89.68 ± 0.14	63.00 ± 0.54	56.00 ± 1.52	61.57 ± 2.92	57.50 ± 1.19	87.16 ± 0.41	82.94 ± 0.28	-
Multi-relational	**RGCN**	87.98 ± 0.14	62.96 ± 0.44	57.60 ± 1.78	65.88 ± 1.86	59.22 ± 1.70	88.01 ± 0.47	82.02 ± 0.23	-
	+ RoSE (8b)	$\underline{90.23 \pm 0.10}$	67.77 ± 0.60	61.40 ± 2.06	71.96 ± 1.82	70.78 ± 1.45	90.28 ± 0.45	86.81 ± 0.16	+ 5.08
	+ RoSE (70b)	89.68 ± 0.14	**71.57 ± 0.42**	63.80 ± 1.86	73.53 ± 1.42	75.31 ± 1.48	**91.77 ± 0.38**	**88.52 ± 0.19**	**+ 7.22**
	HAN	88.68 ± 0.15	63.24 ± 0.54	56.00 ± 1.67	68.82 ± 2.12	58.28 ± 1.99	87.55 ± 0.37	83.32 ± 0.26	-
	+ RoSE (8b)	90.09 ± 0.15	66.83 ± 0.48	60.00 ± 1.47	72.94 ± 1.64	72.50 ± 1.78	89.23 ± 0.28	86.12 ± 0.15	+ 4.55
	+ RoSE (70b)	89.77 ± 0.12	69.55 ± 0.43	62.80 ± 1.86	72.94 ± 1.58	74.38 ± 1.49	90.31 ± 0.38	87.49 ± 0.15	+ 5.91
	SeHGNN	87.97 ± 0.19	62.72 ± 0.52	60.00 ± 1.30	71.37 ± 1.28	65.31 ± 1.95	86.58 ± 0.39	82.53 ± 0.19	-
	+ RoSE (8b)	89.93 ± 0.18	68.27 ± 0.51	62.00 ± 1.41	73.33 ± 1.86	77.34 ± 1.04	89.53 ± 0.32	86.94 ± 0.18	+ 4.41
	+ RoSE (70b)	89.50 ± 0.23	$\underline{70.99 \pm 0.44}$	64.60 ± 2.12	**77.45 ± 1.15**	76.09 ± 1.31	$\underline{91.38 \pm 0.50}$	$\underline{87.96 \pm 0.20}$	$\underline{+ 5.93}$
Edge-featured	**UniMP**	89.92 ± 0.16	69.98 ± 0.58	63.40 ± 1.79	71.18 ± 2.00	78.44 ± 1.50	87.20 ± 0.59	84.29 ± 0.23	-
	+ RoSE (8b)	90.21 ± 0.12	69.55 ± 0.62	$\underline{67.80 \pm 2.13}$	76.08 ± 1.79	**80.94 ± 1.12**	89.17 ± 0.54	86.33 ± 0.21	+ 2.24
	+ RoSE (70b)	**90.37 ± 0.18**	70.41 ± 0.64	$\underline{67.80 \pm 1.78}$	$\underline{76.47 \pm 1.73}$	$\underline{79.84 \pm 1.54}$	89.52 ± 0.41	87.69 ± 0.18	+ 2.52
	GIN	89.77 ± 0.15	67.59 ± 0.41	64.60 ± 2.08	68.63 ± 1.73	73.28 ± 2.06	87.05 ± 0.36	83.03 ± 0.21	-
	+ RoSE (8b)	89.68 ± 0.15	68.27 ± 0.69	**68.20 ± 1.48**	74.51 ± 2.13	79.22 ± 1.19	88.55 ± 0.30	83.32 ± 0.29	+ 2.54
	+ RoSE (70b)	89.55 ± 0.15	69.12 ± 0.68	66.20 ± 1.18	72.75 ± 1.45	77.03 ± 2.05	88.93 ± 0.32	84.84 ± 0.17	+ 2.07
	GraphGPS	OOM	66.85 ± 0.48	60.80 ± 1.73	70.20 ± 1.84	74.53 ± 0.77	85.14 ± 0.45	83.05 ± 0.26	-
	+ RoSE (8b)	OOM	67.69 ± 0.56	66.60 ± 1.88	73.14 ± 2.13	76.56 ± 1.90	87.53 ± 0.30	83.48 ± 0.23	+ 2.41
	+ RoSE (70b)	OOM	68.48 ± 0.54	64.00 ± 1.60	72.75 ± 2.24	77.34 ± 1.49	88.10 ± 0.45	85.24 ± 0.17	+ 2.56

4.4 Efficient Relation Type Annotation

When dealing with graphs with dense edges, the number of edges to be annotated significantly increases, which may incur expensive costs when using non-free LLMs as the backbone. To this end, we introduce an efficient node-wise query edge sampling strategy that reduces the number of queries required for LLM-based relation type classification. We assume that neighboring nodes j_1 and j_2 of a node i, which are close in the feature space, are likely to have similar semantic relationships with i. Building upon this intuition, for each node i, we randomly traverse its neighbors and query their relationships until either (i) all kinds of edge types are discovered or (ii) a predefined patience threshold γ for per-node LLM queries is reached. For the remaining unqueried neighbors, we find their closest annotated neighbor and assign the same relation types as the corresponding annotation, akin to a pseudo-labeling approach. This approach can greatly reduce the number of queries associated with LLM-based edge classification, particularly on graphs with dense edges. The overall procedures is detailed in Algorithm 1. We illustrate the performance and efficiency of this approach in Appendix B.

Algorithm 1 Efficient Relation Type Annotation

```
Input: Node $i$, Neighborhood $\mathcal{N}_i$
Output: List of relationship labels $\mathbf{L}$

$\mathbf{S_{ng}} \leftarrow []$        # List of encountered neighbors
$\mathbf{S_{lb}} \leftarrow []$        # Labels of encountered edges
$c \leftarrow 0$                       # Initialize patience
for $j$ in $\mathcal{N}_i$ do
    if $(|\text{Set}(\mathbf{S_{lb}})| \geq R)$ or $(c \geq \gamma)$ then
        # Upon satisfying (i) or (ii), escape
        break
    else
        Add $j$ to $\mathbf{S_{ng}}$
        Add $\mathcal{M}_c\left(\{\boldsymbol{s}_c\} \cup \{t_i, t_j\}\right)$ to $\mathbf{S_{lb}}$
        $c \leftarrow c + 1$
    end if
end for

# Initialize with labels of encountered edges
$\mathbf{L} \leftarrow \mathbf{S_{lb}}$
for $u$ in $\mathcal{N}_i \setminus \text{Set}(\mathbf{S_{ng}})$ do
    $l \leftarrow \text{argmin}_{v \in \{0,1,\ldots,|\mathbf{S_n}|\}} (\text{dist}(\mathbf{S_{ng}}[v], u))$
    Add $\mathbf{S_{lb}}[l]$ to $\mathbf{L}$
end for
```

5 Experiments

In our experiments, we evaluate our proposed framework on the node classification task using seven well-established benchmarks: Cora [6], Pubmed [7], WikiCS [9], IMDB [31], Cornell, Texas, and Wisconsin [10]. To assess the effectiveness of our approach, we compare **RoSE** with a wide range of existing GNN architectures, including both traditional and popular GNNs [33; 34; 35; 21; 22; 18], as well as transformer-based GNNs [19; 20; 23]. The GNNs considered in our experiments can be broadly broadly categorized as (1) Multi-relational GNNs, such as RGCN [21], HAN [22], and

Table 3: Node classification accuracy (%) on IMDB, Texas, and Cora with multi-relational and edge-featured GNNs, averaged over 10 runs (± SEM). The best and second best performances for each architecture are represented by **bold** and underline.

Multi-relational GNNs		IMDB	Texas	Cora
RGCN	Random	62.90 ± 0.50	66.47 ± 1.67	87.00 ± 0.29
	Distance	66.99 ± 0.48	66.67 ± 2.15	88.03 ± 0.46
	RoSE (8b)	67.77 ± 0.60	71.96 ± 1.82	90.28 ± 0.45
	RoSE (70b)	**71.57 ± 0.42**	73.53 ± 1.42	**91.77 ± 0.38**
	G.T.	68.66 ± 0.57	**76.47 ± 1.82**	-
HAN	Random	62.76 ± 0.59	67.65 ± 1.85	86.19 ± 0.42
	Distance	66.66 ± 0.50	68.63 ± 2.09	87.13 ± 0.49
	RoSE (8b)	66.83 ± 0.48	**72.94 ± 1.64**	89.23 ± 0.28
	RoSE (70b)	**69.55 ± 0.43**	**72.94 ± 1.58**	**90.31 ± 0.38**
	G.T.	68.39 ± 0.62	71.37 ± 2.24	-
SeHGNN	Random	62.46 ± 0.56	70.98 ± 2.09	86.00 ± 0.36
	Distance	67.97 ± 0.43	71.57 ± 1.15	87.07 ± 0.32
	RoSE (8b)	68.27 ± 0.51	73.33 ± 1.86	89.53 ± 0.32
	RoSE (70b)	**70.99 ± 0.44**	77.45 ± 1.15	**91.38 ± 0.50**
	G.T.	69.00 ± 0.48	**78.04 ± 1.07**	-

Edge-featured GNNs		IMDB	Texas	Cora
UniMP	Random	68.65 ± 0.40	71.18 ± 1.90	87.02 ± 0.30
	Distance	69.12 ± 0.68	72.94 ± 1.88	87.94 ± 0.41
	RoSE (8b)	69.55 ± 0.62	76.08 ± 1.79	89.17 ± 0.54
	RoSE (70b)	**70.41 ± 0.64**	**76.47 ± 1.73**	**89.52 ± 0.41**
	G.T.	69.87 ± 0.57	77.84 ± 1.94	-
GIN	Random	67.23 ± 0.42	69.22 ± 1.90	79.96 ± 0.93
	Distance	68.27 ± 0.37	70.59 ± 1.96	86.92 ± 0.50
	RoSE (8b)	68.27 ± 0.69	**74.51 ± 2.13**	88.55 ± 0.30
	RoSE (70b)	**69.12 ± 0.68**	72.75 ± 1.45	**88.93 ± 0.32**
	G.T.	68.54 ± 0.43	74.12 ± 1.59	-
GraphGPS	Random	67.23 ± 0.44	69.41 ± 2.15	85.80 ± 0.25
	Distance	66.98 ± 0.75	69.22 ± 1.76	86.46 ± 0.44
	RoSE (8b)	67.69 ± 0.56	**73.14 ± 2.13**	87.53 ± 0.30
	RoSE (70b)	**68.48 ± 0.54**	72.75 ± 2.24	**88.10 ± 0.45**
	G.T.	67.07 ± 0.78	72.75 ± 1.70	-

SeHGNN [23]; (2) Edge-featured GNNs, including GIN [18], UniMP [19], and GraphGPS [20]; and (3) Single-type edge processing GNNs, such as GCN [33], GAT [34], and JKNet [35]. For the edge decomposition in our framework, we adopted LLaMA3-8b and 70b [26] as foundational LLMs. Detailed dataset descriptions and experimental configurations are specified in Appendix C.

5.1 Main Results

Table 2 presents the node classification accuracy results of integrating various GNN architectures with our proposed **RoSE**, across various datasets. The experiments demonstrate that our method achieves marked improvements in accuracy across multi-relational GNN architectures. Notably, lightweight architectures such as RGCN and HAN, when integrated with **RoSE**, achieve performance comparable to complex transformer-based architectures like UniMP and GraphGPS. For instance, on the WikiCS dataset, RGCN with **RoSE** surpasses the vanilla UniMP architecture, setting a new state-of-the-art performance. Edge-featured architectures also exhibit significant improvements, with gains of up to 6% on Texas and Wisconsin datasets with GIN.

It is worth emphasizing that the integration of **RoSE** consistently enhances performance across all dataset types, regardless of the original accuracy. Particularly impressive improvements are observed on datasets such as IMDB, Cornell, Texas, and Wisconsin, where GNNs have typically struggled. These results underscore the versatility of **RoSE** in improving node classification performance, irrespective of the original dataset composition. Furthermore, the scalability of **RoSE** with larger language models (e.g., **RoSE** 70b) is evident, further boosting performance in most scenarios, highlighting the effectiveness of leveraging advanced reasoning capabilities within the proposed pipeline.

5.2 Additional Experiments

Table 4: Semantic relation types generated from the *relation generator* and filtered from the *relation discriminator*. Short description of each relation is highlighted in **bold** and underline.

Semantic Relations of Cora Dataset	
Retained Relations	**Filtered Relations**
• **Methodology Similarity**: Link papers that utilize similar methodological approaches, algorithms, or architectures to tackle their research objectives. This groups papers based on their technical commonalities.	• **Problem Similarity**: Connect papers that address similar research problems or questions, even if they use different approaches. This captures papers that are thematically related.
• **Contrasting Approaches**: Connect papers that explore divergent or contrasting approaches to a similar problem. This could surface insightful comparisons and foster a more holistic understanding of the problem space.	• **Performance Benchmark**: Associate papers that utilize the same benchmark dataset, evaluation metric, or performance comparison framework. This allows for standardized comparisons across models.
• **Theoretical Foundation**: Link papers that build upon the same fundamental theories, principles or mathematical formulations. This traces the theoretical lineage and underpinnings across papers.	• **Shared Challenges**: Group papers that grapple with similar challenges, limitations or open problems yet to be fully addressed. This synthesizes common hurdles faced by different techniques.
• **Sequential Refinement**: Connect papers where one incrementally improves or optimizes the techniques proposed by the other. This captures the evolutionary trajectory of methods within a research area.	• **Conceptual Parallels**: Link papers that draw conceptual parallels, analogies or inspiration from techniques in other domains and adapt them to the problem at hand. This captures cross-pollination of ideas.
• **Shared Application Domain**: Associate papers that apply their techniques to the same application domain or real-world problem, such as image classification, natural language processing, robotics, etc. This highlights practical use-case similarities.	• **Complementary Insights**: Connect papers that offer complementary insights, where the findings of one augment the understanding or interpretation of the results in another. This provides a more comprehensive picture.

Effect of Relation Discriminator. In this experiment, we analyze the necessity and effectiveness of *relation discriminator*. We begin with a case study on the Cora dataset to demonstrate its necessity. Then, we perform an ablation study on node classification performance on Cora and Texas datasets with and without *relation discriminator* to exhibit its effectiveness.

Table 4 presents the set of retained and excluded relation types from the Cora co-citation dataset, where nodes represent scientific publications with paper abstracts as their text attribute. The relations curated from *relation generator* are generally plausible; however, some generated types are either difficult to determine through textual analysis of node attributes or exhibit significant overlap with each other. For instance, the relation type Performance Benchmark (second relation in the rightmost column) is not easily identified based on paper abstracts, as these abstracts often do not enumerate each benchmark used within the paper. Thus, determining such relations exceeds the capability of language models. Additionally, Complementary Insights (last element of the filtered relations) overlaps significantly with Contrasting Approaches, introducing redundancy. Consequently, such relations are filtered out by the *relation discriminator*. Further case study on Texas dataset is provided in Appendix B.

Table 5: Step-wise evaluation on Texas and Cora in comparison without *relation discriminator*, averaged over 10 runs (± SEM). The best and second-best performances are represented by **bold** and underline.

GNNs		LLaMA3 8b		LLaMA3 70b		
		Texas	Cora	Texas	Cora	Avg Gain
RGCN	w/o $\mathcal{M}_d$	70.00 ± 2.27	87.66 ± 0.42	73.14 ± 1.39	87.94 ± 0.42	
	RoSE	71.96 ± 1.82	**90.28 ± 0.45**	73.53 ± 1.42	**91.77 ± 0.38**	+ 2.20
HAN	w/o $\mathcal{M}_d$	71.37 ± 1.47	86.23 ± 0.31	71.57 ± 1.69	86.52 ± 0.40	
	RoSE	72.94 ± 1.64	89.23 ± 0.28	72.94 ± 1.58	90.31 ± 0.38	+ 2.43
SeHGNN	w/o $\mathcal{M}_d$	72.54 ± 1.49	86.15 ± 0.47	74.51 ± 1.92	86.98 ± 0.38	
	RoSE	73.33 ± 1.86	89.53 ± 0.32	**77.06 ± 0.68**	91.38 ± 0.50	+ 2.78
UniMP	w/o $\mathcal{M}_d$	73.92 ± 2.59	87.55 ± 0.49	75.10 ± 1.67	87.40 ± 0.50	
	RoSE	**76.08 ± 1.79**	89.17 ± 0.54	76.47 ± 1.73	89.52 ± 0.41	+ 1.82
GIN	w/o $\mathcal{M}_d$	70.59 ± 2.20	86.85 ± 0.41	69.61 ± 1.58	86.52 ± 0.41	
	RoSE	74.51 ± 2.13	88.55 ± 0.30	72.75 ± 1.45	88.93 ± 0.32	**+ 2.79**
GraphGPS	w/o $\mathcal{M}_d$	73.33 ± 1.65	85.76 ± 0.19	70.39 ± 2.90	86.72 ± 0.50	
	RoSE	73.14 ± 2.13	87.53 ± 0.30	72.75 ± 2.24	88.10 ± 0.45	+ 1.33

We also empirically validate the efficacy of this filtration on the Texas and Cora datasets by evaluating the node classification performance with and without the *relation discriminator*, as shown in Table 5. Consistent improvements are observed with *relation discriminator* across 23 out of 24 settings, showing an average 2.23% increase in accuracy.

Effect of Relation Decomposer. Table 3 compares the performance of **RoSE** with rule-based decomposition methods on the IMDB, Texas, and Cora datasets. The baselines are formulated as follows: (1) **Random**, which randomly decomposes edges into different relations; (2) **Distance**, which decomposes edges into two relations based on the cosine distance between the associated node features obtained from pre-trained language models (PLMs), categorizing them as semantically similar or different edges. The ground-truth decomposition (**GT**) obtained through manual annotation is also presented for comparison. It is important to note that the ground-truth decomposition consists of mutually exclusive relations, and for the Cora dataset, ground truth information is not available. The results demonstrate the superior performance of **RoSE** compared to basic rule-based methods, highlighting the necessity of leveraging LLMs for intricate semantic decomposition. Moreover, **RoSE** achieves the best or second-best performance on all ablative datasets, even when compared to the ground truth decomposition. This underscores the effectiveness of our *relation decomposer* component, which identifies all relations that accurately describe a given edge, thereby providing a richer source of information for GNN architectures to exploit.

Sensitivity to LLM Temperature. Figure 2 compares the performance of **RoSE** with respect to the decoding temperature. Higher temperature results in higher randomness in the outputs of LLMs, and may influence the performance of the *relation decomposer*. We choose two representative GNN architectures for our evaluation, RGCN from multi-relational GNNs and GIN from edge-featured GNNs. Our experiments on IMDB, Texas, and Cora reveal that the improvements of **RoSE** are consistent across varying temperatures.

6 Related Works

Node Feature-level Enhancement. The presence of textual content in TAGs has inspired researchers to explore beyond traditional feature encoding methods such as bag-of-words [36] and skip-grams [37]. Consequently, numerous studies have been proposed to generate semantically rich node features by employing relatively smaller pretrained language models (PLMs) [1; 11; 12; 13],

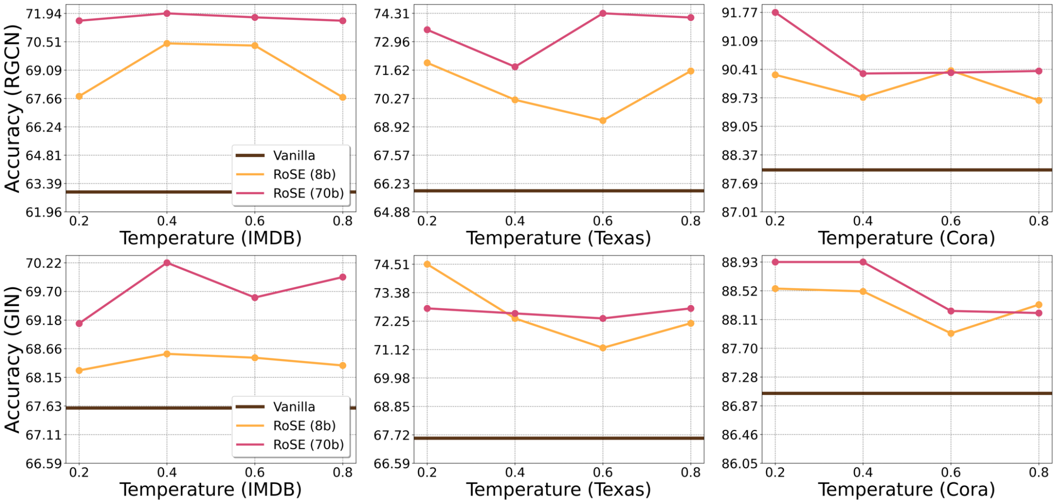


Figure 2: Sensitivity to temperature when prompting *relation decomposer*. Varied temperature (0.2 - 0.8) is denoted on the x-axis, while node classification accuracy(%) is denoted on the y-axis. Red, yellow and brown each denote **RoSE** (LLaMA3-70b), **RoSE** (LLaMA3-8b), and vanilla GNNs (RGCN and GIN), respectively.

including DeBERTa [38], Sentence-BERT [16], E5 [39], and OpenAI's text-ada-embedding-002 [40], alongside larger LLMs such as GPT [24] and LLaMA [26]. These efforts can be broadly categorized into three approaches: **(1) Cascading structure** receives initial node features from the output embeddings of PLMs and LLMs, followed by the deployment of GNNs to obtain final representations. This independent framework has been widely adopted across various studies in TAG literature [2; 4; 41; 42; 3; 11; 14; 43]. **(2) Co-training structure** involves the joint training of PLMs and GNNs within an interactive workflow. This facilitates a dynamic and correlated workflow of semantic information across connected nodes [1; 12; 13]. **(3) Enhanced text augmentation** focuses on enriching the raw textual contents with PLMs and LLMs, such as by replacing text attributes with textual explanations generated by LLMs during its node classification [17] or augmenting external knowledge within a knowledge graph [44; 45]. However, these studies often overlook the diverse semantics inherent in graph structures and characterize edges as a binary adjacency matrix of uniform relation, thus leading to structural oversimplification.

LLMs with Graph Structural Information. Another line of research investigates the potential of LLMs for addressing graph problems by injecting graph structural information into the input prompt of LLMs. This incorporation is achieved through various methods, including describing node adjacency in natural language [46; 47; 48; 49], utilizing syntax tree into natural language representations [50], and leveraging structural tokens [51]. Although these approaches integrate structural data into LLMs, they treat graph edges as binary connections, presenting a clear distinction from our work of utilizing LLMs to automatically decompose graph structures into multiple semantic relation types.

7 Conclusion

Given the limitation of existing TAG literature in simplifying the entangled semantics in graph structure, we introduced **RoSE**, an innovative framework that leverages the analytical capabilities of LLMs to disentangle edges in a fully automated manner, based on the textual contents of connected nodes. As a pioneering effort in revealing and addressing the structural oversimplification, we believe our contributions provide valuable insights into this field. However, one limitation of our framework is its reliance on the general knowledge of LLMs for identifying relation types, which may not fully capture domain-specific relationships when applied to graphs from highly specialized domains that are not well-represented in the LLMs' training data. As future work, we plan to explore techniques such as retrieval-augmented generation (RAG) to effectively incorporate domain knowledge.

References

[1] J. Yang, Z. Liu, S. Xiao, C. Li, D. Lian, S. Agrawal, A. Singh, G. Sun, and X. Xie, “Graphformers: Gnn-nested transformers for representation learning on textual graph,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 28 798–28 810, 2021.

[2] J. Zhou, X. Han, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, “Gear: Graph-based evidence aggregating and reasoning for fact verification,” *arXiv preprint arXiv:1908.01843*, 2019.

[3] Z. Liu, C. Xiong, M. Sun, and Z. Liu, “Fine-grained fact verification with kernel graph attention network,” *arXiv preprint arXiv:1910.09796*, 2019.

[4] J. Zhu, Y. Cui, Y. Liu, H. Sun, X. Li, M. Pelger, T. Yang, L. Zhang, R. Zhang, and H. Zhao, “Textgnn: Improving text encoder via graph neural network in sponsored search,” in *Proceedings of the Web Conference 2021*, 2021, pp. 2848–2857.

[5] Q. Li, X. Li, L. Chen, and D. Wu, “Distilling knowledge on text graph for social media attribute inference,” in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022, pp. 2024–2028.

[6] A. K. McCallum, K. Nigam, J. Rennie, and K. Seymore, “Automating the construction of internet portals with machine learning,” *Information Retrieval*, vol. 3, pp. 127–163, 2000.

[7] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, “Collective classification in network data,” *AI magazine*, vol. 29, no. 3, pp. 93–93, 2008.

[8] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec, “Open graph benchmark: Datasets for machine learning on graphs,” *Advances in neural information processing systems*, vol. 33, pp. 22 118–22 133, 2020.

[9] P. Mernyei and C. W.-C. Cangea, “A wikipedia-based benchmark for graph neural networks. arxiv 2020,” *arXiv preprint arXiv:2007.02901*, 2007.

[10] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery, “Learning to extract symbolic knowledge from the world wide web,” *AAAI/IAAI*, vol. 3, no. 3.6, p. 2, 1998.

[11] E. Chien, W.-C. Chang, C.-J. Hsieh, H.-F. Yu, J. Zhang, O. Milenkovic, and I. S. Dhillon, “Node feature extraction by self-supervised multi-scale neighborhood prediction,” *arXiv preprint arXiv:2111.00064*, 2021.

[12] J. Zhao, M. Qu, C. Li, H. Yan, Q. Liu, R. Li, X. Xie, and J. Tang, “Learning on large-scale text-attributed graphs via variational inference,” *arXiv preprint arXiv:2210.14709*, 2022.

[13] T. A. Dinh, J. den Boef, J. Cornelisse, and P. Groth, “E2eg: End-to-end node classification using graph topology and text-based node attributes,” in *2023 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2023, pp. 1084–1091.

[14] K. Duan, Q. Liu, T.-S. Chua, S. Yan, W. T. Ooi, Q. Xie, and J. He, “Simteg: A frustratingly simple approach improves textual graph learning,” *arXiv preprint arXiv:2308.02565*, 2023.

[15] Z. Chen, H. Mao, H. Li, W. Jin, H. Wen, X. Wei, S. Wang, D. Yin, W. Fan, H. Liu *et al.*, “Exploring the potential of large language models (llms) in learning on graphs,” *ACM SIGKDD Explorations Newsletter*, vol. 25, no. 2, pp. 42–61, 2024.

[16] N. Reimers and I. Gurevych, “Sentence-bert: Sentence embeddings using siamese bert-networks,” *arXiv preprint arXiv:1908.10084*, 2019.

[17] X. He, X. Bresson, T. Laurent, A. Perold, Y. LeCun, and B. Hooi, “Harnessing explanations: Llm-to-lm interpreter for enhanced text-attributed graph representation learning,” in *The Twelfth International Conference on Learning Representations*, 2023.

[18] W. Hu, B. Liu, J. Gomes, M. Zitnik, P. Liang, V. Pande, and J. Leskovec, “Strategies for pre-training graph neural networks,” 2020.

[19] Y. Shi, Z. Huang, S. Feng, H. Zhong, W. Wang, and Y. Sun, "Masked label prediction: Unified message passing model for semi-supervised classification," *arXiv preprint arXiv:2009.03509*, 2020.

[20] L. Rampášek, M. Galkin, V. P. Dwivedi, A. T. Luu, G. Wolf, and D. Beaini, "Recipe for a general, powerful, scalable graph transformer," *Advances in Neural Information Processing Systems*, vol. 35, pp. 14 501–14 515, 2022.

[21] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *The semantic web: 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, proceedings 15*. Springer, 2018, pp. 593–607.

[22] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous graph attention network," in *The world wide web conference*, 2019, pp. 2022–2032.

[23] X. Yang, M. Yan, S. Pan, X. Ye, and D. Fan, "Simple and efficient heterogeneous graph neural network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023, pp. 10 816–10 824.

[24] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[25] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *Advances in neural information processing systems*, vol. 35, pp. 27 730–27 744, 2022.

[26] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[27] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann *et al.*, "Palm: Scaling language modeling with pathways," *Journal of Machine Learning Research*, vol. 24, no. 240, pp. 1–113, 2023.

[28] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," *Advances in neural information processing systems*, vol. 35, pp. 22 199–22 213, 2022.

[29] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.

[30] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–35, 2023.

[31] X. Fu, J. Zhang, Z. Meng, and I. King, "Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding," in *Proceedings of the web conference 2020*, 2020, pp. 2331–2341.

[32] S. Yun, M. Jeong, R. Kim, J. Kang, and H. J. Kim, "Graph transformer networks," *Advances in neural information processing systems*, vol. 32, 2019.

[33] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[34] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[35] K. Xu, C. Li, Y. Tian, T. Sonobe, K.-i. Kawarabayashi, and S. Jegelka, "Representation learning on graphs with jumping knowledge networks," in *International conference on machine learning*. PMLR, 2018, pp. 5453–5462.

[36] Z. S. Harris, “Distributional structure,” *Word*, vol. 10, no. 2-3, pp. 146–162, 1954.

[37] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” *Advances in neural information processing systems*, vol. 26, 2013.

[38] P. He, X. Liu, J. Gao, and W. Chen, “Deberta: Decoding-enhanced bert with disentangled attention,” *arXiv preprint arXiv:2006.03654*, 2020.

[39] L. Wang, N. Yang, X. Huang, B. Jiao, L. Yang, D. Jiang, R. Majumder, and F. Wei, “Text embeddings by weakly-supervised contrastive pre-training,” *arXiv preprint arXiv:2212.03533*, 2022.

[40] A. Neelakantan, T. Xu, R. Puri, A. Radford, J. M. Han, J. Tworek, Q. Yuan, N. Tezak, J. W. Kim, C. Hallacy *et al.*, “Text and code embeddings by contrastive pre-training,” *arXiv preprint arXiv:2201.10005*, 2022.

[41] C. Li, B. Pang, Y. Liu, H. Sun, Z. Liu, X. Xie, T. Yang, Y. Cui, L. Zhang, and Q. Zhang, “Adsgnn: Behavior-graph augmented relevance modeling in sponsored search,” in *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, 2021, pp. 223–232.

[42] Z. Hu, Y. Dong, K. Wang, K.-W. Chang, and Y. Sun, “Gpt-gnn: Generative pre-training of graph neural networks,” in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 1857–1867.

[43] H. Liu, J. Feng, L. Kong, N. Liang, D. Tao, Y. Chen, and M. Zhang, “One for all: Towards training one graph model for all classification tasks,” in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=4IT2pgc9v6

[44] Y. Sun, S. Wang, Y. Li, S. Feng, X. Chen, H. Zhang, X. Tian, D. Zhu, H. Tian, and H. Wu, “Ernie: Enhanced representation through knowledge integration,” *arXiv preprint arXiv:1904.09223*, 2019.

[45] W. Liu, P. Zhou, Z. Zhao, Z. Wang, Q. Ju, H. Deng, and P. Wang, “K-bert: Enabling language representation with knowledge graph,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, pp. 2901–2908.

[46] R. Ye, C. Zhang, R. Wang, S. Xu, and Y. Zhang, “Natural language is all a graph needs,” *arXiv preprint arXiv:2308.07134*, 2023.

[47] J. Guo, L. Du, and H. Liu, “Gpt4graph: Can large language models understand graph structured data? an empirical evaluation and benchmarking,” *arXiv preprint arXiv:2305.15066*, 2023.

[48] H. Wang, S. Feng, T. He, Z. Tan, X. Han, and Y. Tsvetkov, “Can language models solve graph problems in natural language?” *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[49] B. Fatemi, J. Halcrow, and B. Perozzi, “Talk like a graph: Encoding graphs for large language models,” in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=IuXR1CCrSi

[50] J. Zhao, L. Zhuo, Y. Shen, M. Qu, K. Liu, M. Bronstein, Z. Zhu, and J. Tang, “Graphtext: Graph reasoning in text space,” *arXiv preprint arXiv:2310.01089*, 2023.

[51] J. Tang, Y. Yang, W. Wei, L. Shi, L. Su, S. Cheng, D. Yin, and C. Huang, “Graphgpt: Graph instruction tuning for large language models,” *arXiv preprint arXiv:2310.13023*, 2023.

[52] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32*, 2019, pp. 8024–8035.

[53] M. Fey and J. E. Lenssen, “Fast graph representation learning with PyTorch Geometric,” in *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

Supplementary Materials

A Detailed Prompt Templates

In this section, we provide the fixed prompt templates used in our experiments for the *relation generator*, *discriminator*, and *decomposer*.

First, we supply the *relation generator* with detailed information about the graph composition and task description, enabling it to generate a set of candidate semantic relation types. The prompt template for the *generator* is as follows:

> **# Graph Composition Description**
> You are tasked with analyzing a graph... [Graph description]
>
> **# Task Description**
> Your objective is to design a set of unique semantic edge types that capture meaningful relationships between the nodes based on their text attributes.
> Focus on revealing semantic connections that captures unique patterns between specific nodes. These edge types should be inferred from the summarized textual content.
>
> Create edge types as many as you feel are absolutely necessary to decompose, while maintaining a manageable number of edge types for practical decomposition.

Subsequently, we concatenate the relation types curated from *generator* with the task description of edge type filtering, and feed the combined prompt into the *relation discriminator*. The prompt template for the *discriminator* is detailed below:

> **# Task Description**
> You are tasked with verifying the quality and relevance of proposed semantic edge types in a graph representing [Graph description]. Your objective is to identify and retain only the essential edge types for improving the performance of Graph Neural Networks (GNNs) in node classification tasks.
>
> **# Task Requirements**
> When discriminating the edge types, consider the following guidelines: [Requirements]
>
> **# Proposed Semantic Edge Types**
> [Relation types curated from the relation generator]

During the semantic edge decomposition phase, we query the *relation decomposer* to determine all possible relations that the given edge can be categorized under. To accomplish this, we concatenate the instruction prompt with the text attributes of the associated nodes in the input prompt for the *relation decomposer*. The input prompt template is provided as follows:

> **# Task Description**
> You are an helpful assistant, that classifies an edge connection between two nodes into one or more of the following relation types. Note that it is a multiple-choice classification.
>
> **# Relation Specification**
> Relation types are as follows: [List of relation types]
>
> **Node 1**: [Raw text attribute of Node 1], **Node 2**: [Raw text attribute of Node 2]
> **Question**: Carefully choose relation types that likely represent the semantic relation between the two nodes.

B Further Analysis and Experiments

B.1 Additional Case Study

In extension from Section 5, we present the retained and filtered relation types for Texas datasets in Table 6. In the Texas dataset, the Studies_Under/Has_Student Edge is identified as nearly redundant with the Advised_By/Advises Edge, leading to its exclusion to avoid redundancy. Additionally, the Affiliated_With Edge

Table 6: Semantic relation types generated from the *relation generator* and filtered from the *relation discriminator*. Short description of each relation is highlighted in **bold** and underline.

Semantic Relations of Texas Dataset	
Retained Relations	**Filtered Relations**
• **Teaches/Teaches_Under Edge**: Connects a faculty node and a course node (faculty teaches that course). • **Researches/Research_Contributes_To Edge**: Connects a faculty or student node with a project node (they conduct research related to that project). • **Advised_By/Advises Edge**: Connects a student node and a faculty node (faculty advises or mentors that student). • **Enrolled_In/Enrolls Edge**: Connects a student node and a course node (student is enrolled in that course). • **TA_For/Has_TA Edge**: Connects a student node and a course node (student is a teaching assistant for that course).	• **Studies_Under/Has_Student Edge**: Connects a student node to a faculty node suggesting that the student studies under that professor's guidance, without an explicit advising relationship stated. • **Staff_Supports/Supported_By_Staff Edge**: Connects a staff node to other nodes (faculty/student/course/project) implying that the staff provides some type of administrative or technical support for that entity. • **Affiliated_With Edge**: Connects faculty/student/staff nodes to their primary associated entity like a lab, center, department or institute mentioned in their text.

Table 7: Node classification accuracy (%) on various datasets and GNN architectures with efficient querying technique of **RoSE**, averaged over 10 runs (± SEM). The best performance in each architecture is represented by **bold**.

GNN Architectures		IMDB	WikiCS
RGCN	Vanilla	62.96 ± 0.44	82.02 ± 0.23
	RoSE-efficient (8b)	67.22 ± 0.33	86.42 ± 0.18
	RoSE-original (8b)	**67.77 ± 0.60**	**86.81 ± 0.16**
HAN	Vanilla	63.24 ± 0.54	83.32 ± 0.26
	RoSE-efficient (8b)	66.52 ± 0.64	85.81 ± 0.21
	RoSE-original (8b)	**66.83 ± 0.48**	**86.12 ± 0.15**
SeHGNN	Vanilla	62.72 ± 0.52	82.53 ± 0.19
	RoSE-efficient (8b)	66.31 ± 0.37	86.16 ± 0.20
	RoSE-original (8b)	**68.27 ± 0.51**	**86.94 ± 0.18**
UniMP	Vanilla	**69.98 ± 0.58**	84.29 ± 0.23
	RoSE-efficient (8b)	69.36 ± 0.52	86.09 ± 0.19
	RoSE-original (8b)	69.55 ± 0.62	**86.33 ± 0.21**
GIN	Vanilla	67.59 ± 0.41	83.03 ± 0.21
	RoSE-efficient (8b)	67.15 ± 0.56	**84.20 ± 0.28**
	RoSE-original (8b)	**68.27 ± 0.69**	83.32 ± 0.29
GraphGPS	Vanilla	66.85 ± 0.48	83.05 ± 0.26
	RoSE-efficient (8b)	67.41 ± 0.73	**85.14 ± 0.18**
	RoSE-original (8b)	**67.69 ± 0.56**	83.48 ± 0.23

is deemed too ambiguous, as it can encompass various edges generated from the Texas dataset, and is therefore removed. Hence, these findings demonstrate the effectiveness of the *relation discriminator* in identifying and filtering out relations that lack feasibility or distinctiveness, ensuring the retention of meaningful and non-redundant edges.

B.2 Experiments on Efficient Relation Type Annotation

To demonstrate the efficacy of the proposed efficient query edge sampling strategy discussed in Section 4.4, we conduct further experiments with **RoSE** using our efficient relation type annotation (denoted as **RoSE**-efficient) on graphs with the largest number of edges: WikiCS [9] and IMDB [31]. Table 7 displays the node classification performance of multi-relational and edge-featured GNNs, utilizing LLaMa3-8b [26] as a base LLM. As demonstrated in Table 7, **RoSE**-efficient can still improve the performance of original GNNs across 10 out of 12 settings, with less than half the number of queries than **RoSE**-original. Notably, it even surpasses the performance of **RoSE** with full edge annotation (**RoSE**-original) when incorporated with GIN [18] and GraphGPS [20].

Table 8: Comparison of the number of queries sent to *relation-decomposer* by **RoSE** versus **RoSE** with the efficient query technique.

Methods	IMDB	WikiCS
RoSE-efficient (8b)	**15391**	**40055**
RoSE-original (8b)	45698	215603
Decrement	**61.58%↓**	**78.80%↓**

To verify the efficiency of our sampling strategy, we compare the total number of queries sent to the *relation decomposer* by **RoSE** and **RoSE**-efficient. Remarkably, our method reduces the number of queries by more than half, while maintaining comparable performance.

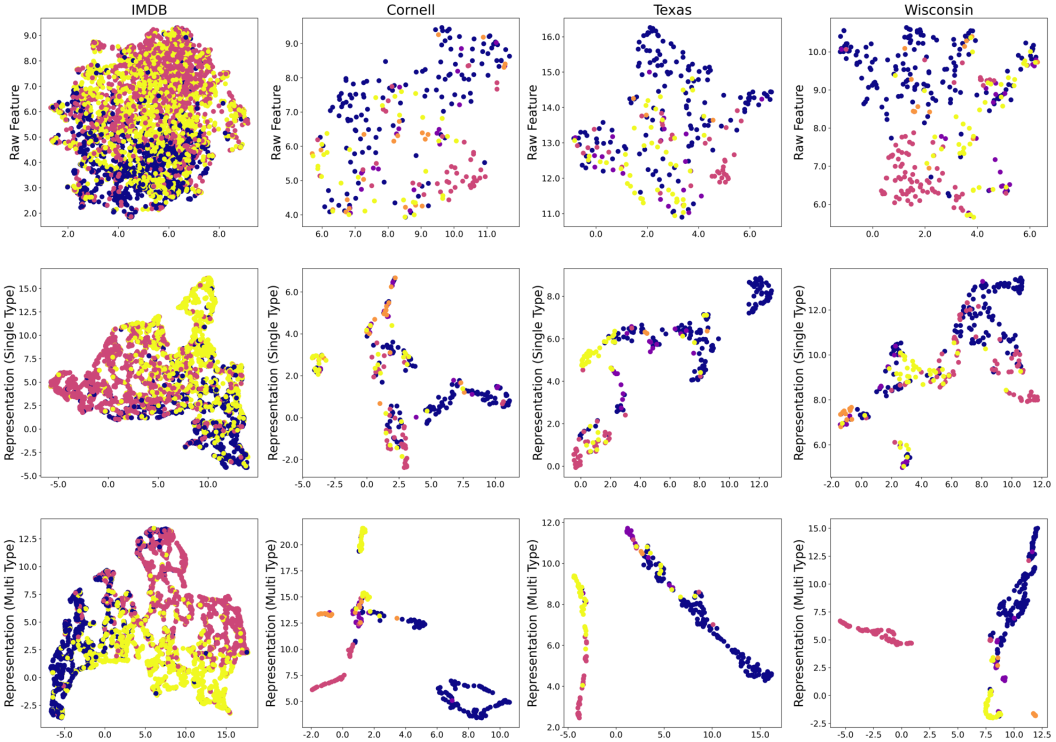


Figure 3: UMAP visualization analysis between raw features and representations of RGCN trained with single and multiple types of relations.

B.3 Importance of Semantic Edge Decomposition - Representational Analysis

We further analyze the enhancements provided by edge-decomposition strategy(presented in Section 3), in a representation learning perspective. Specifically, we analyze the UMAP visualizations of node representations obtained from RGCN [21] and HAN [22], trained with single and multiple types of relations. Figures 3 and 4 illustrate these visualizations, each rows representing: (1) initial node features, (2) node representations learned from RGCN, and (3) node representations learned from HAN, respectively. The results demonstrate that decomposing conventional edges into multiple relation types yields more distinct, clustered representations. Conversely, simplifying the inherent and diverse semantics leads to less distinguishable representations, particularly on the WebKB datasets (Cornell, Texas, and Wisconsin) [10] when using RGCN as the backbone.

We observe similar trends with respect to the inter-prototype similarity between representation prototypes. Specifically, we calculate per-class prototype vector $\mathbf{p}_k = \frac{1}{|C_k|}\sum_{i \in C_k} \boldsymbol{z}_i$, where C_k denotes the set of nodes belonging to class k. Then we evaluate the average cosine similarity between class prototypes as $\texttt{Sim}_{\text{mean}} = \mathbb{E}_{k_1 \neq k_2, \{k_1, k_2\} \subseteq C}\left(\frac{\mathbf{p}_{k_1} \cdot \mathbf{p}_{k_2}}{\|\mathbf{p}_{k_1}\| \|\mathbf{p}_{k_2}\|}\right)$, with C denoting the set of class labels. Intuitively, a smaller $\texttt{Sim}_{\text{mean}}$ implies more distinct class prototypes within the feature space. We plot the $\texttt{Sim}_{\text{mean}}$ along the y-axis of Figure 5. As evident in the figure, our results indicate that simplifying diverse edge semantics results in less distinguishable class representations (i.e. high similarity between class prototypes). This is particularly pronounced in RGCN on Cornell and Texas dataset, where $\texttt{Sim}_{\text{mean}}$ of learned representations on a single relation type is higher than inter-prototype similarities of raw features. In contrast, disentangling these semantics into multiple edge types can achieve significant improvements in inter-class separation. Specifically, for the Cornell dataset, $\texttt{Sim}_{\text{mean}}$ of multi-relation type processing achieves a reduction in similarity of at least 43% across all GNNs, compared to those obtained from raw features and uniform edge type processing.

C Experimental Settings

C.1 Dataset Statistics

In this section, we provide an overview of the graph compositional information for our benchmark datasets:

[1] https://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-11/www/wwkb/

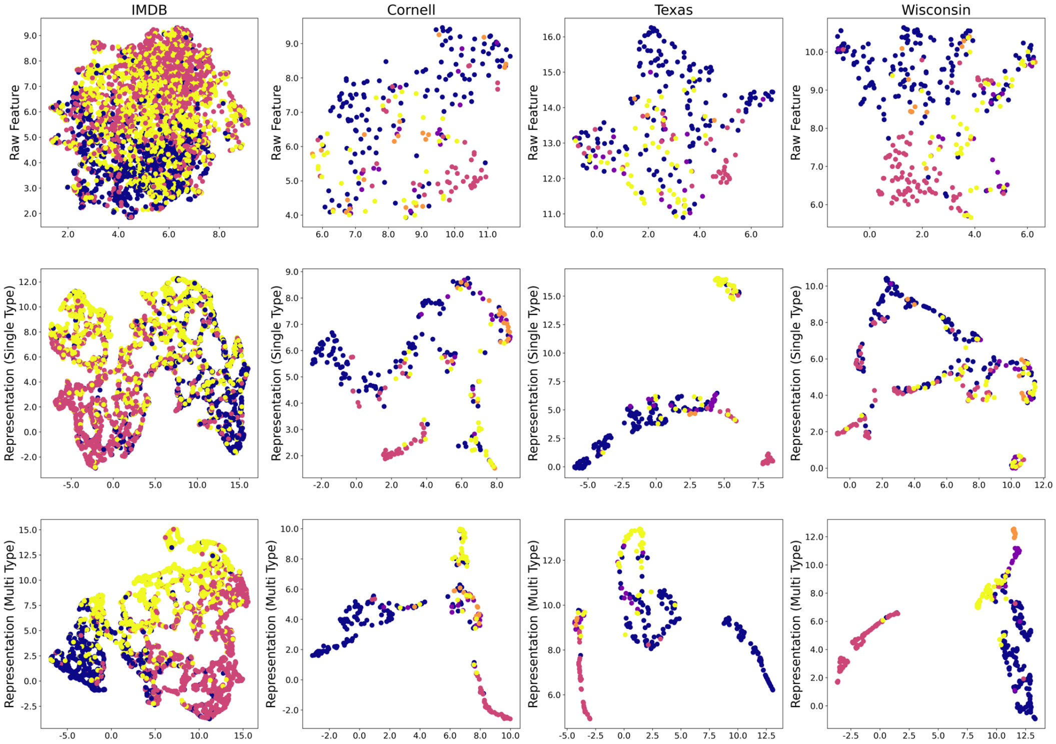


Figure 4: UMAP visualization analysis between raw features and representations of HAN trained with single and multiple types of relations.

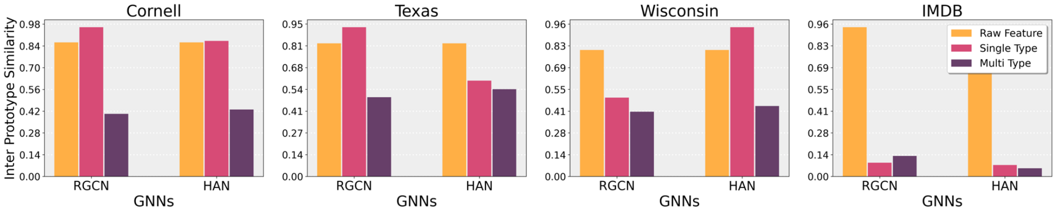


Figure 5: Comparison of average inter-prototype similarity (i.e., average cosine similarity between per-class mean representation vectors) between raw features and representations of GNNs trained with single and multiple types of relations.

Pubmed [7] is a co-citation network in which nodes represent scientific publications and edges denote co-citations. The textual content of each node comprises the paper's abstract. The predefined categories are Diabetes Experimental, Diabetes Type I, and Diabetes Type II.

IMDB [31] is a movie graph where nodes represent movies and edges indicate the overlap of movie professionals. The textual content of each node corresponds to the summarized movie description. The predefined genres are Action, Comedy, and Drama.

WebKB[1] (Cornell, Texas, Wisconsin) [10] are hyperlink networks in which nodes represent web pages and edges are hyperlinks. The text attribute of each node represents the web page content. The predefined categories are Student, Faculty, Staff, Course, and Project.

Cora [6] is a co-citation network where nodes represent scientific papers and edges indicate co-citations. The textual content of each node comprises the paper's abstract. The predefined categories are Case-based, Genetic algorithms, Neural networks, Probabilistic methods, Reinforcement learning, Rule learning, and Theory.

WikiCS [10] is a hyperlink network in which nodes represent web pages and edges are hyperlinks. The text attribute of each node represents the web page content. The predefined categories are Computational linguistics, Databases, Operating systems, Computer architecture, Computer security, Internet protocols, Computer file systems, Distributed computing architecture, Web technology, and Programming language topics.

Table 9: Statistics of TAG benchmark datasets.

Dataset	Pubmed	IMDB	Cornell	Texas	Wisconsin	Cora	WikiCS
#Nodes	19,717	4,182	247	255	320	2,708	11,701
#Edges	44,338	47,789	213	119	449	5,278	216,123
#Classes	3	3	5	5	5	7	10
Domain	Citation	Movie	Hyperlinks	Hyperlinks	Hyperlinks	Citation	Hyperlinks

Comprehensive statistics of the datasets used in our experiments, including the graph domain and the number of nodes, edges, classes, are provided in Table 9.

C.2 Implementation Details

We adopted Sentence-BERT [16] to encode node features and relational features when using edge-featured GNNs. To carefully identify qualified relation types, we employ Claude Opus[2] (Chat version) from Anthropic as the *relation generator* and *discriminator*. The edge decomposition is performed using a LLaMA3 [26]-based *relation decomposer*, which is a free, open-sourced model. In our experiments, we utilize LLaMA3-8b and 70b as base LLMs, with a fixed temperature of 0.2 across all settings. Adhering to the same evaluation protocols of existing TAG works [15; 17], we adopt the same train/validation/test splits of 60%/20%/20%, respectively. For training the GNN models, all architectures are implemented using PyTorch [52] and PyTorch Geometric [53]. All experiments are conducted on RTX Titan and RTX 3090 (24GB) GPU machines. Throughout all experiments, we set the hidden dimension to 64 and employ the Adam optimizer with a weight decay of 0. The best validation performance is selected within the following hyperparameter search space:

- Learning rate: $[0.001, 0.005, 0.05, 0.01]$
- Number of layers: $[2, 3]$
- Dropout: $[0, 0.1, 0.5, 0.8]$

D Broader Impacts

Our work identifies a novel bottleneck in GNN performance for downstream tasks, specifically highlighting the oversimplification of graph structures. To address this, we introduce **RoSE**, a framework that decomposes edges to enhance the representational learning capabilities of GNNs. This shift in focus from node attributes, which dominated prior studies, to the structure itself represents a significant paradigm shift. By leveraging the general knowledge of LLMs, our approach opens new research avenues for improving graph structures. Our analysis demonstrates that **RoSE** significantly enhances classification performance of GNNs, particularly in datasets where GNNs have traditionally underperformed. Consequently, our work extends the applicability of GNN architectures to a broader spectrum of datasets, overcoming previous performance limitations and expanding their utility in various domains.

[2] https://www.anthropic.com/claude

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We claim that over-simplification of edges hinder downstream performance of GNN on node classifciation tasks, which is analyzed in Section 3. Motivated by this, we utilize LLMs to automize the process, described in Section 4, and empirically evaluated on Section 5.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We state our limitations on Section 7, our method may shortfall in scenarios where excessive domain expertise is required, for LLMs may general knowledge of a specific domain.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (*e.g.*, independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, *e.g.*, if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: Instead of a theoretical analysis, we provide an empirical one, illustrated in Section 3, highlighting the importance and necessity of decomposing predefined edge types.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.

- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide detailed settings required in our experiments in Appendix C, reporting all necessary implementation specifications such as hyperparameter search space, datasets, etc..

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (*e.g.*, in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (*e.g.*, a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (*e.g.*, with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (*e.g.*, to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All the models and datasets we used are openly available, as specified in Section 5 and Appendix C. We will release our code upon rebuttal period.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (*e.g.*, for a new open-source benchmark).

- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (*e.g.*, data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide detailed settings required in our experiments in Appendix C, reporting all necessary implementation specifications such as hyperparameter search space, datasets, etc..

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Upon all tables, we provide standard errors upon training and evaluating GNNs. For LLM querying, we use existing open-source LLMs, which do not involve training.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (*e.g.*, Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (*e.g.*, negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide information regarding hardware specs employed in our experiments in Appendix C

   Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (*e.g.*, preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: Our paper adheres to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (*e.g.*, if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: We provide a Broader Impact section in Appendix D

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (*e.g.*, disinformation, generating fake profiles, surveillance), fairness considerations (*e.g.*, deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (*e.g.*, gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (*e.g.*, pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our paper does not pose such risks. We do not release any pretrained language models, image generators, nor scraped datasets.

    Guidelines:

    - The answer NA means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (*e.g.*, code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have included citations for all the benchmark datasets, LLMs and GNN architectures within our paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (*e.g.*, CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (*e.g.*, website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: Our paper does not release any new assets; our work is conducted upon existing LLM models, benchmark datasets, and GNN architectures.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our paper does not involve any crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our paper does not involve any crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.